

OBJECT ORIENTED COMMUNICATIONS SYSTEM OVER THE INTERNET

The present application claims the benefit of U.S. provisional patent application 60/208,045, filed on May 31, 2000, and 60/242,708, filed on October 25, 2000, as well as United States application Serial No. 09/867,469, filed on May 31, 2001.

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates to a software and hardware infrastructure allowing for the management of any network device regardless of its location utilizing only a single communications protocol.

DESCRIPTION OF THE PRIOR ART

Although a browser can be used to directly request images, video, sound etc. from a server, more usually, an HTML document converts the presentation of information served to the browser by a server. However, generally the contents of an HTML file are static in that the browser can only present a passive snapshot of the contents at the time the document is served. In order to present dynamic information, such as information generated by an application or device or to obtain from the user data which has been inserted into an HTML-generated form, conventional world wide web service employ a "raw" interface such as a common gateway interface (CGI). The HTML file provides no mechanism for presenting dynamic information generated by an application or device, except through the CGI.

With respect to obtaining data from a user for use by the application or device, although standard HTML provides a set of tags which implements a convenient mechanism for serving interactive forms to the browser, complete with text fields, check boxes and pull down menus, the CGI must be used to process submitted forms. Form processing is important to remote control, management, configuration, monitoring and diagnosing applications because forms processing is a convenient way to configure an

application according to a user input utilizing the world wide web communications model. Unfortunately, form processing employing a CGI is extremely complex, requiring an application designer to learn and implement an unfamiliar interface. Therefore, a CGI is not a suitable interface for rapid development and prototyping of graphical user interfaces (GUI).

Furthermore, a developer must then master a native application source code language (such as C and C++), HTML and the CGI, in order to develop a complete application along with its user interface. Additionally, the CGI mechanism assumes access to the application on the customer's site, but not to the object. The person or process that invokes the CGI program must have full knowledge of that program. A firewall is not in control of the CGI. Therefore, the use of the CGI would make it impossible to fully secure communications between users. Finally, the CGI mechanism works only in the client server environment communicating to one server. This method cannot be applied to the distributed multi-process environment, if only a single communications protocol for the entire system is to be utilized.

Other systems could be used instead of the CGI. For example, an object model such as Microsoft's DCOM or the object management groups internet inter-ORB protocol (IIOP) or the common object request broker architecture (CORBA) could be employed. However, these technologies have some limitations when it comes to creating web services. For example, DCOM and IIOP/CORBA are rich environments, which means that implementations and applications that use them tend to be complex and symmetrical. In other words, to build a distributed application using them, one typically must require the same distributed object model running at both ends of the connection. However, the internet does not guarantee the specific kind of client or server software which would be running at the second end of the connection. All that is required is that the connection understands the hypertext transfer protocol (HTTP).

Additionally, it is often technically impossible to insure that all applications would run either IIOP or DCOM.

Any server connected to the internet can potentially be accessed by an internet user, which raises some obvious security problems. To address these concerns, most organizations insert a firewall between their publically accessible web servers and the masses that can access the servers. Generally, a firewall can block incoming traffic based on various criteria and thereby increase an organization's confidence in the security of its system. While they are essential to the secure use of the internet, firewalls make the efficient use of distributed object protocols very challenging. Understanding why firewalls cause problems for distributed object protocols requires understanding of how a firewall is able to distinguish one protocol from another. For example, in the TCP/IP architecture, each widely used protocol is assigned its own port number and each request made using this protocol carries that number. HTTP, for example, is generally assigned port 80, while, for example, the file transfer protocol (FTP) relies on port 21. Most firewalls allow blocking a specific protocol by rejecting all traffic sent on the port used by that protocol. In general, firewalls are configured to allow traffic on port 80. If this was not the case, HTTP requests from browsers could not be received. However, many firewalls block most other ports. This would result in other protocols not being received by the user's local server.

Unlike HTTP, FTP and other widely used protocols, distributed object protocols do not generally have a single well known port number assigned to them. Instead, these protocols typically use dynamically assigned ports, with port numbers chosen arbitrarily as needed. If no firewall intervenes in the communication between the client and the server, this approach works fairly well. If a firewall is asserted, communication stops because the firewall blocks all traffic using this protocol since it is not configured to pass requests on arbitrary port numbers.

One response to this challenge is to use existing internet standards such as an HTTP and XML. More than any other application protocol, the HTTP connects most users to one another. Millions of web sites and browsers utilize protocol. The problem with HTTP alone is that it is mainly a mechanism for passing files from a server to a client. To create more ambitious web servers, the HTTP must be expanded.

One manner of extending the HTTP would be to use an object protocol such as the simple object access protocol (SOAP) copyrighted by IBM, Lotus Development Corporation, Microsoft and User Land Software. This protocol adds a set over the HTTP headers and rich XML payload to enable complex application-to-application communication over the internet. In other words, the SOAP messaging protocol uses HTTP to carry messages that are formatted with XML.

SUMMARY OF THE INVENTION

The deficiencies of the prior art are addressed by the present invention which is directed to a system utilizing an object access protocol as the single communications protocol between the user, any intermediate software system, and a network element (hardware). The invention utilizes the same object protocol, not only as the foundation of the internet communication, but also for the lower level inter-object communication. The system not only serves as a remote control for a network server across a firewall, but can also be used to control the switches running the network from the other side of the firewall. Because the server can utilize the simple network management protocol (SNMP) to configure the switches, the present invention would be able to control a network's back-end remotely, and will even be capable of changing bandwidth assignments. Because of its object access nature, the present invention is adaptable to control any network device be it a server, a switch, or any number of other devices such as network printers and fax machines. A host of emerging "smart" products including computer

controlled appliances such as refrigerators and other kitchen appliances, climate control systems and medical devices can also be controlled.

5 The present invention is cross platform compatible since it is capable of running regardless of the platform because it is built in industry and standard protocols and is therefore not restricted to any given platform, thereby making a completely unique product.

10 The present invention will consist of a client and a number of servers. The client end would include a web browser interface enabling the utilization of HTTP compatibility with the SOAP which is designed specifically enable the transmission of XML payloads with special HTTP headers.

15 Every server of the present invention will run a parser filtering out the XML messages and will translate these messages into necessary commands to be executed behind the firewall. The XML vocabulary will be translated into the necessary SNMP commands to be issued to the switches behind the firewall. The present invention will be scalable, thereby allowing for the
20 addition of management capabilities by the integration of additional command protocols for communicating with other devices. The present invention will initially control devices that use the SNMP or Command Line Interface (CLI) protocol and can be scaled to add support for future protocols and devices as
25 these become industry standards.

The architecture of the present invention allows a user to wrap existing or new communications equipment in a SOAP/XML wrapper, which would then allow for internet readiness or manageability via the internet anywhere in the world. The
30 software wrapper is the computer program that understands object access protocol, i.e. contains a Protocol Virtual Machine (PVM), and runs on a single board computer, or on the hardware of the network element. A T-Box has a direct (not via the Internet) connection to the controlled network elements. The T-Box can
35 communicate to the network elements via the Simple Network

Management Protocol (SNMP) or Comment Line Interface (CLS). Additionally, a network management system would be utilized with the SOAP/XML protocol as the foundation for inter-object communication.

5 The SOAP protocol extends the XML into the remote procedure call (RPC) paradigm in a standardized manner. The present invention extends the simple RPC paradigm to a more complex useful transaction-based approach whereby a "call" by user could be chained through a series of servers each providing a telecommunications management network (TMN) hierarchy management level over the internet. This web chaining of the SOAP requests is a unique enhancement.

10 The SOAP infrastructure is important to the technology of the present invention. Available SOAP parsers had to be broken up and modified into client and server side parsing engines. Along with the new parsers, the purposes of asynchronous identifications, a connection pooling mechanism was developed upon which the SOAP messages rode. The utilization of this system would allow network elements to be remotely controlled beyond a firewall.

15 Other and further advantages, objects and features of the invention will become apparent to those skilled in the art from a reading of the following detailed description of a preferred embodiment of the invention when taken in light of the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

20 **Figure 1** is a block diagram showing the hardware of the present invention as well as an upward ordered message flow;

25 **Figure 2** is a block diagram of the present invention as well as showing a downward ordered message flow;

30 **Figure 3** is a diagram showing an alarm path flow;

Figure 4 is a diagram showing alarm object communication and distribution; and

Figure 5 is a diagram showing an alternate embodiment of the present invention.

100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287

DETAILED DESCRIPTION OF THE PRESENT INVENTION

The system design **10** of the present invention is shown in Figures 1 and 2. As shown therein, separate CPUs **12** and **14** would include portions of the system hardware of the present invention. Note that the application can have as many CPUs as servers. We have shown two CPUs only for clarity reasons. The CPU **12** would include a web browser **16** associated with a GUI which is an applet loaded by the web browser from the localized web server **20**. This would prompt the user for data to perform a particular command. Once the required data is entered, the command is encoded into an HTTP-SOAP message. Although the drawings illustrate the utilization of two CPUs, the present invention can employ as many CPUs as there are servers. The use of only two CPUs in the present invention was for clarity purposes only.

In this invention, the implementation of the SOAP message is similar to SNMP format. It recognizes the Managed Information Base (MIB) tree as the representation of hardware component structure. The difference is that in existing implementations of SNMP messages are usually numeric. For example,

```
> Name-of-SNMP-program 10.3 set 23.21 11.11.2.12 2
```

The legend is: Name-of-SNMP-program node_address command (get\set) class.item object_id value.

Since SOAP is an XML based textual protocol, instead of numerals, it contains meaningful strings representing class, item, and object id in the component hierarchy. Within the SOAP envelope, the SNMP-like set of commands can be expanded if required.

Additionally, unlike most SNMP implementations, the present invention always guarantees the delivery of the message, referred to as "nailing connections". The receiving server in our system always notifies the sender.

The second CPU **14** would include a web server (Apache) which is the entry point of the network management system of the present invention. The web server **20** would determine if the message is a SOAP message. If this is the case, it would strip off the HTTP header and forward the remaining SOAP/XML message to the SOAP server **22**. This server is a servlet container which, upon receiving a SOAP message, a determination would be made to which server the message is destined. In this instance, the message will always be routed from or to a read/write server (RWS) **24**. Upon receipt of the message from the SOAP server **22**, the RWS **24** will store the data in a generic model in a relational database **30** in an object to relational table mapping. The RWS **24** will then forward or receive the message from a network management agent (NMA) **26**.

The NMA **26** will then determine the course of action required for the particular type of request or command and build the appropriate model for it using G.855, G.85X recommendations as its basis. The NMA **26** will then forward commands to or receive comments from various network element agents (NEA) **28**. Although a single NEA **28** is shown in Figures 1 and 2, it is noted that a number of NEAs can be employed. The NEA **28**, upon receiving a nodal command from the NMA **26** will build a M.3100 management model to support the nodal action. The NEA **28** will then forward a smaller command to a translation device (T-box) **32** for processing or translation. The T-box **32** then translates the SOAP command into the native language of the intended device such as network element **34**, or SNMP.

The web browser **16** will generally be a third party provided browser. Most personal computers come with a pre-installed browser or can easily be downloaded. Both Netscape and Internet Explorer will be supported as both are Java enabled. However, it is also possible to use either front end applications written in any other computer language, as long as they are talk SOAP/XML. The web browser **16** will upload applets from the web

server **18** for procedural contacts and communicate the network data transactions back to the web server **20** to be forwarded to the RWS **24**. When a user is selecting a network element or set of network elements to perform an action, a network element record must already have been installed into the database **30** for that network element. The user will only be allowed to select from the list of network elements currently being managed. The web browser will also receive asynchronous events from the web server and display them.

The RWS **24** will receive or send user requests and store them in the database **30** as they are received. After a successful storage, a SOAP envelope will be sent to the NMA **26** to begin execution of the user request. The RWS **24** will also forward alarms to the web server **20** for display when it receives them from lower layer processors such as an NEA **28**.

The NMA **26**, when triggered by the RWS **24** will initiate processing of the user request. The NMA **26** will read and parse the SOAP request and then build a model required to satisfy that request. The NMA **26** will send a series of SOAP encoded singular node access requests to the NEA each with the data sufficient to satisfy the secondary request. Thus, a typical NMA network request is broken down into a set of simpler nodal requests. The NMA will implement a standard space model in order to satisfy the network management layer requirement. The NMA will process the request to either a successful or error termination condition. The NMA will log the cumulative status of the user request and also send a response SOAP packet back to the RWS **24** to be forwarded to the user. The NMA **26** will store the data required for each nodal requested to the database **30**. Each NMA request and secondary NEA request all will have unique transaction Ids and database tables.

A Network Element Discovery server (NED) **25** can be provided between the NMA **26**, the RWS **24** and the NEA **28** included in Figures 1 and 2. The NED will discover and store in the

database **30** each network element's configuration. The configuration data may contain port, card, slot and shelf information. It may also discover the current set of provisional connections and their states.

5 The NED **25** will use the NEA **28** for access to the network elements in order to perform its discovery task. The NED will be used to populate the database tables initially. It will also perform an Audit/Learn function.

10 It is noted that a firewall **36** separates the web browser **16**, the web server **20** and the SOAP server **22** from the remaining elements of the system.

15 The NEA **28** will be triggered from a SOAP envelope received from the NMA **26**. The NEA will parse the XML document which encompasses the data necessary to complete a single nodal transaction. An M.3100 derivative model will be used to satisfy the elemental management layer requirement. The NEA **28** will proceed to encode the appropriate SOAP packets to be delivered to one or more of the T-boxes **32** which are managing the network elements to where the request is destined. Although Figures 1 and 2 show the T-box **32** connected to a single network element **34**,
20 it can be appreciated that a single T-box can manage several network elements. To explain, one T-box can manage several network elements. When a success or failure message is received, the status is then stored in the database for logging purposes and then the status is sent back to the NMA **26**.
25

30 The purpose of the database **30** is of course to store information. Once a user hits the appropriate button on the browser **16**, an NMA request will be received by the RWS **24** and immediately stored in the database. Additionally, each NEA request is stored in the database before being processed by the NEA **28**. Furthermore, the configuration of each network element is also stored in the database **30**.

 The T-box **32** contains embedded software. The T-box will be able to receive "M-GET", "M-SET" and "M-ACTION" requests

via the SOAP interface, as well as command line interfaces (CLI). The T-box will translate the SOAP command into the appropriate native console command (MML), (CLI) or SNMP, and send it via an ethernet interface to the actual network element in order to satisfy the given NEA request.

The HTTP-SOAP protocol provides the present invention a method of information exchange between functional components. Similar to any messaging protocol, it provides for data encapsulation with messages. The HTTP-SOAP protocol includes an envelope defining a framework for describing what is in a message and how to process it. It also includes a set of encoding rules for expressing instances of application defined data types as well as a convention for representing remote procedure calls and responses.

It is important to note that the T-box can control the network element from the applet directly to the T-box without control from additional servers. The applet is a JAVA program that runs within the web browser. However, any application written in any language can talk to the network element through the T-box using SOAP as one protocol.

An upward flow of data from a network element **34** to the web browser **16** in the first CPU **12** will now be described. The network element **34** sends a message to the T-box **32** over, perhaps, the ethernet as shown by Step 1. The T-box **32** includes a parser therein and utilizes the parser to translate the response received from the network element **34** into a SOAP response and sends this response to the NEA **28** at Step 2. It is noted that a plurality of NEAs would be utilized and the message relayed from the T-box **32** would be sent to the NEA **28** which is managing that particular network element. At Step 3, the NEA **28** forwards the nodal response created therein to the NMA **26**. The NMA then forwards the entire transaction response to the RWS **24** at Step 4. The RWS **24** will then forward this response to the SOAP server **22** at Step 6, as well as to the database **30** at Step 5. The SOAP

server **22** would then forward the response to the web server **20** which in turn would send the response back to the web browser **16** in the first CPU **12**.

Figure 2 illustrates a downward message flow from the CPU **12** ultimately to a network element **34**. Initially, the web browser **16** will connect to the web server **20** at Step 2. In this case, the web server will automatically upload the applet that will prompt the user for a command and a set of attributes used to control the network element **34**. If authorized by the user, the web browser **16** would then be connected to the web server **20** in the second CPU **14** as shown by Step 2. At this point, the web browser will then build and send an HTTP-SOAP envelope to the web server **20**. Upon receipt of the SOAP envelope, the web server **20** will immediately route it to the SOAP server **22** as shown by Step 3. Once the SOAP server **22** determines that a proper request has been made, it will be forwarded across the firewall **36** to the RWS server **24** as shown by Step 4. At Step 5, the RWS **24** will modify the request if required and then store it in the database **30**. The RWS server will also forward the request to the NMA **26** as shown by Step 6. The NMA **26** will process this message and forward the necessary subcomponents to the proper NEA **28** for nodal processing. Step 8 indicates that the NEA **28** will forward the appropriate single-space command to the T-box **32**. The T-box **32** will formulate and send the proper translation for the command to the network element **34** across the ethernet as shown by Step 9.

Figures 3 and 4 illustrate the alarm path network of the system of the present invention also indicating that the architecture is very scalable and allows for several NMAs as well as several NEAs and one or more GUIs and T-boxes. As alarms are received by the NEAs, they are immediately forwarded to the NMA for further processing and forwarding. Alarms can be received by the NEA at any time and are discriminated via the header

field. Thus an in-progress provision cannot get confused with alarms being interpreted as responses. The directions of the alarm message will be from the Network Element to the T-Box and then to the NEA. The alarm will then proceed to the NMA, the RWS and the GUI (web browser). The alarms, by nature, go from the hardware equipment to the web browser.

A user that connects to the RWS from a GUI will register with the alarm distributor class for alarms. As alarms are received by the NEA, they are immediately forwarded to the NMAs for further processing and forwarding. Alarms can be received by the NEA at any time and are discriminated via the header field. The header attribute "alarm" indicates to the SOAP senders that this is an alarm data object. The body attributes contained in the rest of the alarm information. These attributes are such as NE name and message text.

The present invention employs a TCP/IP network with a connection to the internet. The network has a firewall between itself and the internet. It is possible for a network administrator to access and configure all equipment on the network through the use of telnet, SNMP, and other network management tools. These tools can only be used from behind the firewall. The firewall will, however, prevent these connections from being made from behind the firewall. With the present invention, it is now possible to HTTP connections which are still left open by the firewall to send in SNMP commands to the equipment beyond the firewall. This would allow the present invention to control the bandwidth being assigned to any part of network. This would also enable ISPs and other providers of bandwidth to sell bandwidth in a more dynamic manner because the bandwidth could be adjusted to meet demands more quickly and easily than currently possible.

The present invention also has the ability to allow a single network administrator to administer networks in different geographic locations. For example, if a service provider is located in Canada, network elements can be connected to a T-box

in Canada utilizing a serial cable, an ethernet cable to a hub. The provider would then send email to a customer located, for example, in the United Kingdom confirming this connection. The customer would then email to an engineering technical support team located, for example, in China so that it can assign the correct IP addresses to the switches through a web interface. After these addresses are quickly assigned, the switches are registered for technical support if debugging will be required. The debugging of the switch can be done in various locations through the web interface. Once the registration is completed, the user will start provisioning of the network element using the web interface. Using one or more of the screens of the GUI, the provider in Canada will make selections relating to various equipment that must be connected. When the T-box in Canada receives a generic command, it will make a decision to translate this to CLI or SNMP for the network element. Most of the debugging commands usually include CLI access.

In a second scenario, several computers and peripheral devices are connected to one switch. It is noted that this switch can be used to enable or disable different ports that are used to connect the different devices together via the switch. Furthermore, the speed of the data that travels through these ports can also be altered.

In this manner, traffic can be regulated through the system. Alternatively, different virtual local area networks can be created and connected to that one switch. This is done for the purposes of isolating data that users can share on the computers or to isolate peripheral devices.

The present invention can be utilized to control "smart" devices such as computerized appliances and the like. In a home computerized environment, equipment can help the homeowner with climate control. The present invention can interface with electrical, wired or wireless devices or equipment. In such a scenario, the present invention can be used to program these devices and help in the automation process. The

present invention is also capable of interfacing with wireless data systems and devices as well as interfacing with medical devices, distant learning equipment and various other telecommunications equipment.

5 The present invention offers a highly distributed network management application by using an XML based communication protocol. The translator for the XML based protocol to SNMP or TL1 can be embedded in a small hardware device serving as a web server controlling a number of network
10 elements.

Figure 5 illustrates an alternate embodiment of the present invention in which a user request is transmitted directly from a browser/application 40 to the T-box 48 and then to the network element or application 50 instead of proceeding directly
15 through a database 42, network manager 44 and element manager 46 as illustrated in Figures 1 and 2. A graphical user interface (GUI) is used to control the communications between the browser/application 40 and the network element/application 50 directly through the T-box 48 or indirectly through the network
20 manager 44 and element manager 46. As shown by the double headed arrows, communication is provided either upwardly or downwardly directly through the T-box 48 or indirectly through the T-box and database 42, network manager 44 and element manager 46.

25 The T-box 48 is provided with both a SOAP server as well as the PVM. The GUI is provided with two input devices on the screen. These input devices would indicate whether the user request is to proceed directly to the T-box 48 or via the network management system shown in Figures 1 and 2. As illustrated in
30 Figure 5, this network management system would include the database 42, the network manager 44 as well as the element manager 46. This is important since if the network management system is not operating, communication can still be provided from the browser application 40 directly to the T-box 48 and then to
35 the network element/application 50. It is important to note that this communication can also proceed from the network application

50 to the T-box 48 and then directly to the browser/application 40. This is true since the T-box 48 can transfer a user request from a native language into SOAP as well as translate the SOAP packet into the native language.

5 The user can also specify whether a native command or generic command would be utilized. Generally, the user would not concern themselves with the command names that were given to a particular vendor. In this case, the user would utilize the generic command set. The GUI would include NE NAME as a pop down field automatically populated with the names of the network elements. For example, these names might correspond to LAB 7 Router, 1548M Cisco switch and UofORouter corresponding to the Cisco 7000 router.

10 The present invention utilizes the following commands: Connect, disconnect, setVlan, showVlan, setPortUp as well as setPortDown.

15 The input data includes parameters that the user can set. For example, in the case of the Virtual Lan (setVlan), the user can specify which port is connected to which Vlan. After making a selection, the user can submit the request. An applet will create the SOAP message and send it through the selected path. Request and reply log area will display the appropriate logs and the user interpretation view will display logs and alarms coming from the equipment.

20 It is noted that Figures 1 and 2 show the use of the present invention sending user requests from a browser/application to a network/application through at least one firewall. Figure 5 indicates that the present application would also operate in a situation in which no firewalls are present. However, it is noted that all of the embodiments can operate with or without firewalls.

25 Whereas the preferred form of the present invention has been shown and described herein, it should be realized that there can be many modifications, substitutions and alterations thereto.